



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Experiment No.2
Mapping ER/EER to Relational schema model.
Date of Performance:
Date of Submission:



Aim :- Prepare the schema for Relational Model with the ER/ERR diagram, drawn for the identified case study in experiment no.1.

Objective :- To map the Entity Relationship (ER) / Extended Entity-Relationship (EER) Diagram to Relational Model schema and learn to incorporate various schema-based constraints.

Theory:

Mapping an Entity-Relationship (ER) model to a relational database schema involves translating the conceptual model represented in the ER diagram into tables and relationships in a relational database management system (DBMS). Here are the general rules for mapping ER to a schema in a DBMS:

1. Entities to Tables:
 - a. Each entity in the ER diagram corresponds to a table in the relational schema.
 - b. The attributes of the entity become the columns of the table.
 - c. The primary key of the entity becomes the primary key of the table.
2. Relationships to Tables:
 - a. Many-to-Many Relationships:
 - i. Convert each many-to-many relationship into a new table.
 - ii. Include foreign key columns in this table to reference the participating entities.
 - iii. The primary key of this table may consist of a combination of the foreign keys from the participating entities.
 - b. One-to-Many and One-to-One Relationships:
 - i. Represented by foreign key columns in one of the participating tables.
 - ii. The table on the "many" side of the relationship includes the foreign key column referencing the table on the "one" side.
 - iii. The foreign key column typically references the primary key of the related table.
3. Attributes to Columns:
 - a. Each attribute of an entity becomes a column in the corresponding table.
 - b. Choose appropriate data types for each attribute based on its domain and constraints.
 - c. Ensure that attributes participating in relationships are represented as foreign keys when needed.
4. Primary and Foreign Keys:
 - a. Identify the primary key(s) of each table based on the primary key(s) of the corresponding entity.
 - b. Ensure referential integrity by defining foreign keys in tables to establish relationships between them.



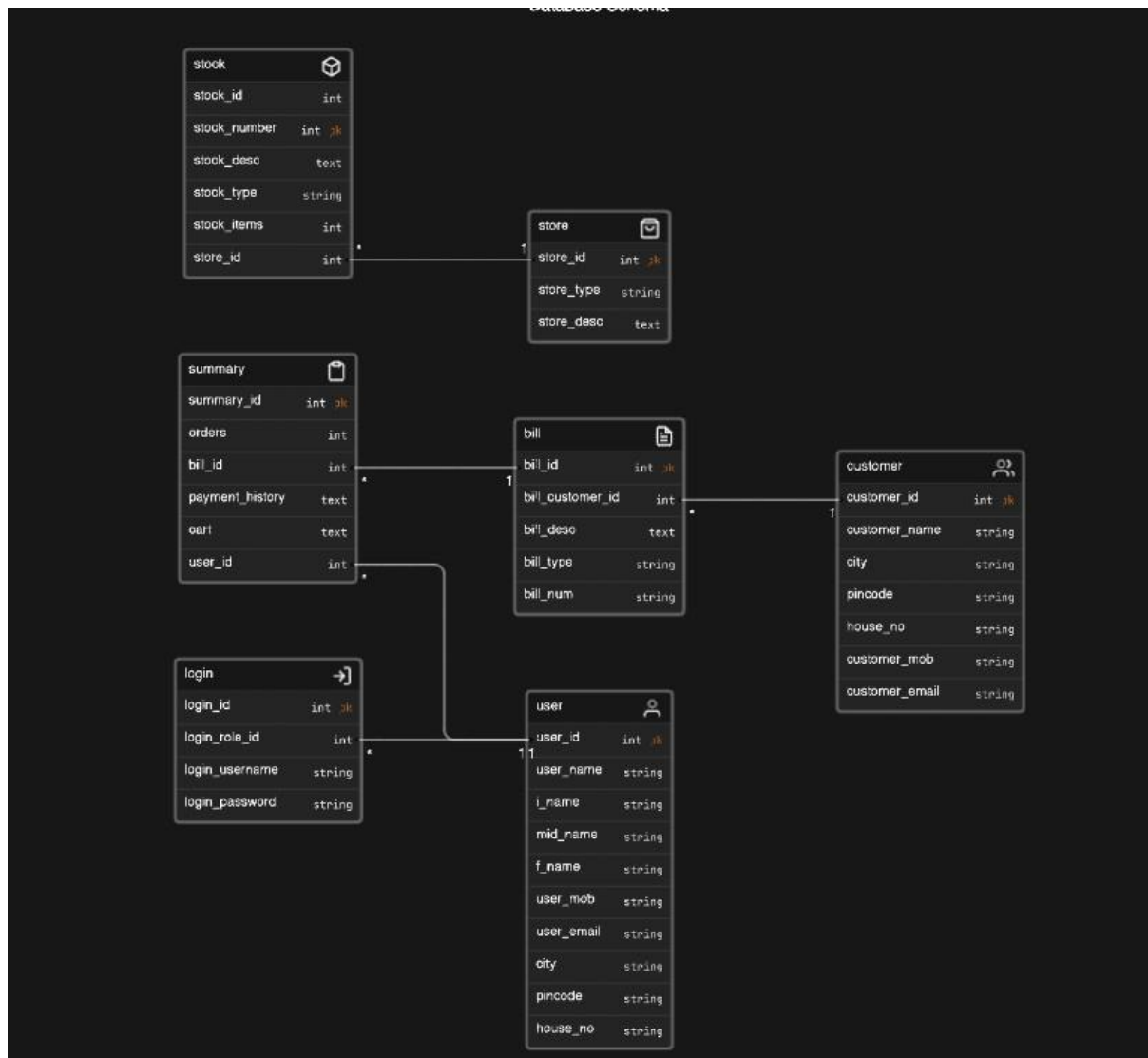
- c. Foreign keys should reference the primary key(s) of related tables.
 - d. Ensure that foreign keys have appropriate constraints, such as ON DELETE CASCADE or ON UPDATE CASCADE, to maintain data integrity.
5. Cardinality Constraints:
- a. Use the cardinality constraints from the ER diagram to determine the multiplicity of relationships in the relational schema.
 - b. Ensure that the constraints are enforced through the appropriate use of primary and foreign keys.
6. Normalization:
- a. Normalize the schema to minimize redundancy and dependency.
 - b. Follow normalization rules such as First Normal Form (1NF), Second Normal Form (2NF), Third Normal Form (3NF), etc., to ensure data integrity and minimize anomalies.
7. Indexing and Optimization:
- a. Consider indexing frequently queried columns to improve query performance.
 - b. Evaluate the schema design for optimization opportunities based on query patterns and performance requirements.

Implementation



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science



Conclusion

- write definition of relational schema and notations
- write various schema-based constraints



In conclusion, relational schema and schema-based constraints play crucial roles in database design and management:

1. Relational Schema:

- Definition: A relational schema defines the structure of a relational database, including tables, columns, and relationships between tables.

- Notations:

- Table: Represented as a rectangle with the table name at the top.

- Attributes: Represented as ovals inside the table rectangle, specifying the attribute name and data type.

- Primary Key: Underlined attribute(s) indicate the primary key(s) of the table.

- Foreign Key: Attributes representing foreign keys establish relationships between tables.

- Example:

Student

student_id (PK): int

name: varchar

age: int

2. Schema-Based Constraints:

- Primary Key Constraint:

- Ensures uniqueness and non-nullity of a column or combination of columns.



- Example: ``PRIMARY KEY (student_id)``

- Foreign Key Constraint:

- Enforces referential integrity by ensuring that values in a column match values in a related table's primary key.

- Example: ``FOREIGN KEY (dept_id) REFERENCES Department(dept_id)``

- Unique Constraint:

- Ensures that values in a column or combination of columns are unique.

- Example: ``UNIQUE (email)``

- Check Constraint:

- Validates data integrity by specifying a condition that must be satisfied for each row.

- Example: ``CHECK (age >= 18)``

- Not Null Constraint:

- Ensures that a column does not accept NULL values.

- Example: ``age INT NOT NULL``

These schema-based constraints ensure data integrity, consistency, and accuracy within the database, enforcing rules and restrictions on the data stored in the tables. By defining a clear relational schema and applying appropriate constraints, database administrators can maintain the quality and reliability of the database system, facilitating efficient data management and retrieval operations.